Optimizing utility: Best Strategy for Answering Queries

CompSci 590.03
Instructor: Ashwin Machanavajjhala
Recap: Laplace Mechanism

**Thm:** If sensitivity of the query is $S$, then adding Laplace noise with parameter $\lambda$ guarantees $\varepsilon$-differential privacy, when

$$\lambda = \frac{S}{\varepsilon}$$

**Sensitivity:** Smallest number s.t. for any $d$, $d'$ differing in one entry,

$$|| q(d) - q(d') || \leq S(q)$$

**Histogram query:** Sensitivity = 2
- Variance / error on each entry = $2\lambda^2 = 2\times4/\varepsilon^2$
Example 1: Enforcing Constraints

• Database of values \{x_1, x_2, ..., x_k\}

• Query Set:
  - Value of \(x_1\) \(\eta_1 = x_1 + \delta_1\)
  - Value of \(x_2\) \(\eta_2 = x_2 + \delta_2\)
  - Value of \(x_1 + x_2\) \(\eta_3 = x_1 + x_2 + \delta_3\)

• But we know that
  - \(\eta_1\) and \(\eta_2\) should sum up to \(\eta_3\)
  - \(\eta_1 \leq \eta_3\) AND \(\eta_2 \leq \eta_3\)
Example 2: Query Strategy

• Query Set:
  – Value of x1
  – Value of x2
  – Value of x1 + x2

• Strategy 1: Answer all queries
  – Sensitivity = 2; Error in each query is $8/\varepsilon^2$.

• Strategy 2: Answer query 1 and query 2.
  Query 3 = query 1 + query 2
  – Sensitivity = 1
  – Error in query 1 and query 2: $2/\varepsilon^2$.

Lecture 13: 590.03 Fall 13
Today’s class

• How to answer a workload of queries with the least error?

• Constrained inference:
  – Ensure the query answers are consistent with each other
  – Order constraints
  – Sum constraints

• Query Strategy:
  – A workload $W$ may be best answered by answering a different query set $A$, and then computing $W$ from $A$
  – Hierarchical, Wavelet and Matrix Mechanism for linear queries.
Outline

• Strategies for answering a query workload

• All Range queries on 1D domain
  – Hierarchical Mechanism [Hay et al VLDB 10]
  – Wavelet Mechanism [Xiao et al ICDE 09]

• General Query Workloads
  – Matrix Mechanism [Li et al PODS 10]
Note

- The following solution ideas are useful whenever
  - You want to answer a set of correlated queries.
  - Queries are based on noisy measurements.
  - Each measurement (x1 or x1+x2) has similar variance.
When do you do this?

• Query workload $W$ has a high sensitivity
  – Removing or changing a tuple affect a large number of the queries

• Query strategy $A$ has low sensitivity

• Each query in $W$ can be answered using a small number of query answers in $A$
  – This ensures that the noise in answering queries in $W$ is not too high
Example Workload: All Range Queries

• Given a set of values \{v_1, v_2, \ldots, v_n\}
• Let \(x_i\) = number of tuples with value \(v_1\).
• Range query: \(q(j,k) = x_j + \ldots + x_k\)

Q: Suppose we want to answer all range queries?
All Range Queries

• Given a set of values \( \{x_1, x_2, \ldots, x_n\} \)
• Range query: \( q(j,k) = x_j + \ldots + x_k \)

Q: Suppose we want to answer all range queries?

Strategy 1: Answer all range queries using Laplace mechanism

• Sensitivity = \( O(n^2) \)
• \( O(n^4/\varepsilon^2) \) total error across all range queries.
• May reduce using constrained optimization ...
All Range Queries

- Given a set of values \{x_1, x_2, ..., x_n\}
- Range query: \( q(j,k) = x_j + ... + x_k \)

Q: Suppose we want to answer all range queries?

Strategy 2: Answer all \( x_i \) queries using Laplace mechanism
Answer range queries using noisy \( x_i \) values.

- \( O(1/\varepsilon^2) \) error for each \( x_i \).
- Error(\( q(1,n) \)) = \( O(n/\varepsilon^2) \)
- Total error on all range queries : \( O(n^3/\varepsilon^2) \)
Hierarchical Mechanism for Range Queries

Strategy 3:
Answer *sufficient statistics* using Laplace mechanism
Answer range queries using noisy sufficient statistics.

[Hay et al VLDB 2010]
Hierarchical Mechanism for Range Queries

- Sensitivity: $\log n$
- $q(2,6) = x_2 + x_{34} + x_{56}$

Error = $2 \times 3\log^2 n/\epsilon^2$
Hierarchical Mechanism for Range Queries

• Every range query can be answered by summing at most $\log n$ different noisy answers
• Maximum error on any range query = $O(\log^3 n / \varepsilon^2)$
• Total error on all range queries = $O(n^2 \log^3 n / \varepsilon^2)$
Summary of Hierarchical Mechanism

• Answering all range queries on a 1D attribute
  – Optimal mechanism for range queries is also an optimal mechanism for computing the CDF.

• Error depends on the query strategy
  – Noisily answer all range queries: Total error = $O(n^4/\varepsilon^2)$
  
  – Noisily answer each count, and use them to answer range queries: Total error = $O(n^3/\varepsilon^2)$
  
  – Hierarchical Mechanism
    Answer counts on a binary tree noisily: Total error = $O(n^2 \log^3 n / \varepsilon^2)$
Outline

• All Range queries on 1D domain
  – Hierarchical Mechanism [Hay et al VLDB 10]
  – Wavelet Mechanism [Xiao et al ICDE 09]

• General Query Workloads
  – Matrix Mechanism [Li et al PODS 10]
Wavelet Mechanism

Step 1: Compute Wavelet coefficients

Step 2: Add noise to coefficients

Step 3: Reconstruct original counts

\[ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ \ldots \ \ldots \ x_n \]

\[ y_1 \ y_2 \ y_3 \ y_4 \ y_5 \ \ldots \ y_n \]

\[ C_1+\eta_1 \ C_2+\eta_2 \ C_3+\eta_3 \ \ldots \ C_m+\eta_m \]
Haar Wavelet

\[
c_0 = \frac{x_1 + x_2 + \cdots + x_n}{n}
\]

\[
\begin{array}{cccccccc}
3 & 2 & 2 & -1 \\
9 & 3 & 6 & 8 & 4 & 5 & 7
\end{array}
\]

\(v_1\) \(v_2\) \(v_3\) \(v_4\) \(v_5\) \(v_6\) \(v_7\) \(v_8\)

\(c_0\) = 5.5 \(c_1\) = -0.5

\(c_2\) \(c_3\)

\(c_4\) \(c_5\) \(c_6\) \(c_7\)
For an internal node,
Let \( a \) = average of leaves in left subtree
Let \( b \) = average of leaves in right subtree

\[
c = \frac{a - b}{2}
\]
Haar Wavelet Reconstruction

Sum of coefficients on root to leaf path
- + if $x_i$ is in the left subtree of coefficient
- - if $x_i$ is in right subtree

$x_4 = c_0 + c_1 - c_2 - c_5$
$x_5 = c_0 - c_1 + c_3 + c_6$
Haar Wavelet : Range Queries

Range Query: number of tuples in a range $S = [a, b]$

Let $\alpha(c)$ be the number of values in the left subtree of $c$ that are in $S$
Let $\beta(c)$ be the number of values in the right subtree of $c$ that are in $S$

$$y = |S| \cdot c_0 + \sum_{c \neq c_0} (c \cdot (\alpha(c) - \beta(c)))$$
Haar Wavelet: Range Queries

\[ y = |S| \cdot c_0 + \sum_{c \neq c_0} (c \cdot (\alpha(c) - \beta(c))) \]

\[ \alpha(c) - \beta(c) = 0 \text{ when no leaves under } c \text{ are contained in } S \]

\[ \alpha(c) - \beta(c) = 0 \text{ when all leaves under } c \text{ are contained in } S \]

Only need to consider those coefficients with partial overlap with the range.
Adding noise to wavelet coefficients

• Associate each coefficient with a weight
  \[
  \text{level}(c) = \text{height of } c \text{ in the tree.}
  \]
  \[
  W_{\text{Haar}}(c) = 2^{h-\text{level}(c)+1}
  \]

• Generalized sensitivity (\(\rho\))
  \[
  \sum_{c \in C} (W(c) \cdot |c(D) - c(D')|) \leq \rho \cdot ||D - D'||_1
  \]
Adding noise to wavelet coefficients

Theorem: Adding noise to a coefficient $c$ from $\text{Laplace}(\lambda/W(c))$ guarantees $(2\rho/\lambda)$-differential privacy.

Proof:

$$\frac{P[M(D) = \langle \tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_k \rangle]}{P[M(D') = \langle \tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_k \rangle]} = \frac{\prod_i e \left( \frac{W(c_i)}{\lambda} \cdot |c_i(D) - \tilde{c}_i| \right)}{\prod_i e \left( \frac{W(c_i)}{\lambda} \cdot |c_i(D') - \tilde{c}_i| \right)}$$

$$\leq e^{\sum_i \left( \frac{W(c_i)}{\lambda} \cdot |c_i(D') - c_i(D)| \right)} \leq e^{\frac{2\rho}{\lambda}}$$
Generalized Sensitivity of Wavelet Mechanism

\[ \rho = 1 + \log_2 n \]

Proof:

• Any coefficient changes by \( 1/m \), where \( m \) is the number of values in its subtree.
• \( m = 1/W(c) \)
• Only \( c_0 \) and the coefficients in one root to leaf path change if some \( x_i \) changes by 1.
Error in answering range queries

• Range query depends on at most $O(\log n)$ coefficients.

• Error in each coefficient is at most $O(\log^2 n/\varepsilon^2)$

• Error in a range query is $O(\log^3 n/\varepsilon^2)$
Summary of Wavelet Mechanism

- Query Strategy: use wavelet coefficients
- Can be computed in linear time
- Noise in each range query: $O(\log^3 n/\varepsilon^2)$
Outline

- All Range queries on 1D domain
  - Hierarchical Mechanism
  - Wavelet Mechanism

- General Query Workloads
  - Matrix Mechanism
Linear Queries

- A set of linear queries can be represented by a matrix
- $\mathbf{X} = [x_1, x_2, x_3, x_4]$ is a vector representing the counts of 4 values
- $\mathbf{H}_4 \mathbf{X}$ represents the following 7 queries
  - $x_1+x_2+x_3+x_4$
  - $x_1+x_2$
  - $x_3+x_4$
  - $x_1$
  - $x_2$
  - $x_3$
  - $x_4$

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
$$

$\mathbf{H}_4$
Query Matrices

Identity

\[
I_4 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Binary Index

\[
H_4 = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Haar Wavelet

\[
Y_4 = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 \\
\end{bmatrix}
\]
Sensitivity of a Query Matrix

- How many queries are affected by a change in a single count?

\[
\text{Sensitivity} = 1 \\
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
\text{Sensitivity} = 3 \\
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
1 & -1 & 0 & 0 \\
0 & 0 & 1 & -1 \\
\end{bmatrix}
\]
Laplace Mechanism

\[ \mathcal{L}(W, x) = Wx + \left( \frac{\Delta W}{\epsilon} \right) \tilde{b}. \]
Matrix Mechanism

\[ \mathbf{y} = \mathcal{L}(\mathbf{A}, \mathbf{x}) \]

\[ \hat{\mathbf{x}}_\mathbf{A} = \mathbf{A}^+ \mathbf{y}, \]

where \( \mathbf{A}^+ = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \) is the pseudo-inverse of \( \mathbf{A} \).
Reconstruction

\[ I_4 \]

(a) \( I_4^{-1} \)

\[ Y_4 \]

(b) \( H_4^t \)

\[ \frac{1}{27} \times \begin{bmatrix}
3 & 5 & -2 & 13 & -8 & -1 & -1 \\
3 & 5 & -2 & 8 & 13 & -1 & -1 \\
3 & -2 & 5 & -1 & -1 & 13 & -8 \\
3 & -2 & 5 & -1 & -1 & -8 & 13 \\
\end{bmatrix} \]
Matrix Mechanism

\[ M_A(W, x) = WA^+ L(A, x). \]

\[ = WA^+ (Ax + \left( \frac{\Delta A}{\epsilon} \right) \tilde{b}) \]

\[ = W \left( x + \left( \frac{\Delta A}{\epsilon} \right) A^+ \tilde{b} \right) \]
Error analysis

$$\text{ERROR}_A(w) = \text{Var}(w\hat{x}_A) = \text{Var}(wx + \left(\frac{\Delta A}{\epsilon}\right)wA^+\tilde{b})$$

$$= \left(\frac{\Delta A}{\epsilon}\right)^2 \text{Var}(wA^+\tilde{b}).$$

$$\text{Var}(wA^+\tilde{b}) = wA^+\text{Var}(\tilde{b})(wA^+)^t$$

$$= wA^+2I_mA^+(wA^+)^t$$

$$= 2w(A^tA)^{-1}A^tA((A^tA)^{-1})^tw^t$$

$$= 2w(A^tA)^{-1}w^t,$$

$$\text{TOTAL ERROR}_A(W) = \left(\frac{2}{\epsilon^2}\right) \Delta_A^2 \text{trace}((A^tA)^{-1}W^tW)$$
Extreme strategies

• Strategy A = $I_n$
  – Noisily answer each $x_i$
  – Answer queries using noisy counts

\[
\text{TOTALERROR}_{I_n}(W) = \left( \frac{2}{\varepsilon^2} \right) \text{trace}(W^tW)
\]

• Strategy A = $W$
  – Add noise to all the query answers

\[
\text{TOTALERROR}_{W}(W) = \left( \frac{2}{\varepsilon^2} \right) \Delta_W^2 n
\]

Good when each query hits a few values.

Good when sensitivity is small (or each value hits a small # queries)
Finding the Optimal Strategy

• Find A that minimizes TotalError_A(W)
  – Reduces to solving a semi-definite program with rank constraints
  – $O(n^6)$ running time.

• See paper for approximations and an interesting discussion on geometry.
Summary

• A linear query workload and strategy can be modeled using matrices

• Previous techniques to find a better strategy to answer a batch of queries is subsumed by the matrix mechanism

• General mechanism to answer queries.

• Noise depends on the sensitivity of the strategy and $A^tA^{-1}$
Next Class

- Sparse Vector Technique
  - Answering a workload of “sparse” queries
References

